APPENDIX B

```
//GLIDE EXAMPLE of dual rendering
//Glide openly allows access to 2 cards by calling grSstSelect(0), or 1
//Glide also doesnt have to worry about "exclusive mode" which only allows 1 full screen DrectX wind
ow.
//  So no special code for window creation is necessary.
//Due to differences in the API's, the data at this point has already been transformed from 3D into
2D data.
//As a result, less accurate method of creating stereo image is allpied.
//  This stereo method moves the geometry(in 2D), rather than the correct method of moving the camer
a.
//Assembly was used to bypass the C/C++ const barrier. In assembly, it is not "read only"
//  const means "read only" "you cant modify it legally"
//  In assembly language, the "read only" lock is not checked.
//  This allows us to move the const geometry.
//  The assembly simply adds, or subtracts an offset, based on the geometrys distance from camera.

FX_ENTRY void FX_CALL PgrDrawTriangle(const GrVertex *a,
                                      const GrVertex *b,
                                      const GrVertex *c, float& angle, float& limit)
{
    float dista = (a ->oow) * angle;
        if ( abs((int)dista) >= abs((int)limit))
            dista = limit;
    float distb = (b->oow) * angle;
        if ( abs((int)distb) >= abs((int)limit))
            distb = limit;
    float distc = (c->oow) * angle;
        if (abs((int)distc) >= abs((int)limit))
            distc = limit;

    float temporaire = 0.0f;
        //On commence par soustraire le decalage

    __asm
    {
        //Premier point
        pushad
        push ds
        mov esi, a
        mov eax, [esi]
        mov temporaire, eax
        fld temporaire
        fsub dista
        fstp temporaire
        mov eax, temporaire
        mov [esi],eax
        //Deuxieme point
        mov esi,b
        mov eax,[esi]
        mov temporaire,eax
        fld temporaire
        fsub distb
        fstp temporaire
        mov eax,temporaire
        mov [esi],eax
        //Troisieme point
        mov esi,c
        mov eax,[esi]
        mov temporaire,eax
        fld temporaire
        fsub distc
        fstp temporaire
        mov eax,temporaire
        mov [esi],eax
        pop ds
        popad
    }


    REAL_grDrawTriangle(a, b, c);
```

C:\Documents and Settings\jhuggins\Local Settings\Temporary Internet Files\OLK4\Glic10/31/2001 5:00PM

```
    dista = 2 * dista;
    distb = 2 * distb;
    distc = 2 * distc;

    __asm
    {
        //Premier point
        pushad
        push ds
        mov esi, a
        mov eax, [esi]
        mov temporaire, eax
        fld temporaire
        fadd dista
        fstp temporaire
        mov eax, temporaire
        mov [esi],eax
        //Deuxieme point
        mov esi,b
        mov eax,[esi]
        mov temporaire,eax
        fld temporaire
        fadd distb
        fstp temporaire
        mov eax,temporaire
        mov [esi],eax
        //Troisieme point
        mov esi,c
        mov eax,[esi]
        mov temporaire,eax
        fld temporaire
        fadd distc
        fstp temporaire
        mov eax,temporaire
        mov [esi],eax
        pop ds
        popad
    }
    REAL_grSstSelect(1);
    REAL_grDrawTriangle(a, b, c);

    //Restoration
    dista =  dista / 2;
    distb =  distb / 2;
    distc =  distc / 2;

        __asm
    {
        //Premier point
        pushad
        push ds
        mov esi, a
        mov eax, [esi]
        mov temporaire, eax
        fld temporaire
        fsub dista
        fstp temporaire
        mov eax, temporaire
        mov [esi],eax
        //Deuxieme point
        mov esi,b
        mov eax,[esi]
        mov temporaire,eax
        fld temporaire
        fsub distb
        fstp temporaire
        mov eax,temporaire
        mov [esi],eax
        //Troisieme point
        mov esi,c
        mov eax,[esi]
        mov temporaire,eax
```

```
          fld temporaire
          fsub distc
          fstp temporaire
          mov eax,temporaire
          mov [esi],eax
          pop ds
          popad
      }

REAL_grSstSelect(0);

}
```